



DECUS

PROGRAM LIBRARY

DECUS NO. 8-554

TITLE ANOVA and DUNCAN

AUTHOR Marjorie H. Kleinman

COMPANY Center for Community Research
New York, New York

DATE June 19, 1972

SOURCE LANGUAGE FORTRAN

DECUS

PROGRAM LIBRARY



1. Title

2. Author

3. Subject

4. Date

5. Location

6. Remarks

ANOVA ANALYSIS OF VARIANCE PROGRAM

DECUS Program Library Write-up

DECUS NO. 8-554

This program can be used for any case in which there are up to four factors, each with up to four levels. However, the total number of treatments must not exceed 64. Missing data is permitted.

Input can be from teletype, paper tape, cards, or magnetic tape. If input is on magnetic tape, it must reside on unit 1 with the file name 'ANOVIN.DA'. The format for the input is completely flexible - the factor values and the score may appear in any order, and the format statement is read in at execution time. All values are read in integer format.

The program will compute and print out for each group the number of subjects, mean, and standard deviation. In addition, T Tests are performed. Then the results of the analysis of variance are printed. In addition, there is an option for calling DUNCAN, the Duncan multiple range program, and passing to it the appropriate information.

MINIMUM HARDWARE

PDP-8/I with 8K of core, with teletype, printer, and disk or DECtape.

OTHER PROGRAMS NEEDED

PS/8 Operating System with FORTRAN Compiler.

NOTE

The program consists of 5 chained segments. The first is named ANOVA and contains subroutine INPUT. The other 4 segments are TSUMS, SUMSQ, SUMIN, and DUNCAN, respectively. Each must be compiled, loaded, and saved separately on the system device. Specify I and O options to the loader for each.

To allow for execution-time formatting of input data, load ANOVA (with subroutine INPUT) and change location 4051 from 4103 to 3144 before saving on the system device. For an explanation of executive-time format statements see the accompanying write-up

DUNCAN Multiple Range Program

This program uses as input information computed by the Analysis of Variance program, and is automatically called by this program if so optioned.

The program works as follows: First, the means for all groups are sorted so that they are in order from lowest to highest. The residual mean square and degrees of freedom associated with it are also passed to DUNCAN by the Analysis of Variance Program, so that the standard error of a single mean can be computed. Then the shortest significant ranges for the differences between means can be determined, using special tables of significant studentized ranges (.01 and .05 level) that have been developed for this multiple range test.

Each difference between means is significant if it exceeds the corresponding shortest significant range. It will be marked with an asterisk in the output.

For information on the running of this program, see the write-up for ANOVA, Analysis of Variance program.

HOW TO RUN ANOVA ANALYSIS OF VARIANCE PROGRAM (AND DUNCAN)

- 1) .R ANOVA (Starting Address is 02403).
- 2) Type in no. of factors (max. is 4). Do not include score.
- 3) Type in no. of levels for each factor (total no. of treatments must not exceed 64). Format is (4I1).
- 4) Type in format statement (may be up to 72 characters in length).
- 5) If Duncan multiple range program is wanted, type a 1 in response to this question. Otherwise, press return.
- 6) Type which variable is the score. If it is first, type 1. If it is last to be read in, and there are 4 factors, type 5, etc.
- 7) Type in a heading to appear on output. It may be as long as 72 characters.
- 8) Type in the input device (1,2,3, or 4).*
- 9) If DUNCAN program is optioned, type in level of table wanted (1 or 5), type in the no. of treatment groups, and the no. of subjects per group, in answer to these questions. Type each on a separate line. The number of subjects in each group must be the same.

Note: The address of the Read statement is 4044. If a card reader error occurs, stop, load this address and press start.

* If input is from magnetic tape, it must reside on unit 1 with the file name 'ANOVIN.DA'.

Execution-time Format Statements

Execution-time format statements are not permitted by the compiler. However, by changing one location in the program after it is in core, it is possible to read in a format statement (from the teletype only), and use it for reading the input. Set up the program (or subprogram) as follows:

```

      DIMENSION FMT (12)
C      This is the array where the format statement will
C      be stored.
      READ (1,15) FMT
15     FORMAT (12A6)
C      A format statement as long as 72 characters can
C      be read in here.
      :
      :
      :
      :
      :
20     READ (IN,25) DATA1,DATA2
C      Input data is read in from device IN
25     FORMAT (F3.0)
C      This is a dummy format that will not be used.
      :
      :
      :
      :
      :
      END
```

Be sure to obtain an assembly listing of the program, and also a map at execution time, so that the precise location to be changed can be determined.

Load the program and all subroutines, and begin execution. When the program pauses or waits to accept data, the appropriate location can be changed.

Now look at the assembly listing. Determine the actual address of the first word of array FMT (See page C-4 of 8K SABR Assembler manual or PS/8 manual for computing actual addresses.)

Now look through the listing for the FORTRAN statement where the reading of the data is done, in this case, statement 20. You will notice that in the assembled code will appear the following:

4033	CALL	2, READ	(call to subroutine READ with 2 arguments)
6201	ARG	(n	(n is the device number)
DDDD			(DDDD is the address where the device number is stored.)
6201	ARG	/fa	(fa is the format number, in this case, 25)
FFFF			(FFFF is the address of the beginning of the format statement.)

In our example, FFFF is the address of the first word of format statement 25. This is the location to be changed, so that it contains the address of the first word of array FMT. Therefore, compute the address where FFFF appears and change its contents accordingly. (As a check on your computations, first compute the actual value FFFF, then load the address where FFFF appears, and see if the contents agree with your computation of FFFF. Then reload the address and deposit the address for the first word of FMT.)

Once this one location has been changed, save the core image on tape.

Altering Format Statements Already in Core

Format statements are stored 2 characters per word in packed, 6-bit ASCII code. Thus, the first half of the first word of a format statement is always a 50, the stripped ASCII code for open parenthesis. F2.0 takes 2 locations in storage, and appears as 0662 for F2, and 5660 for .0. Therefore, one can change all or any part of a format statement in core by determining its exact location and substituting the codes for the desired format. For any format statement that is likely to be changed in this manner, it is advisable to leave plenty of extra spaces when writing the program.

